ROS Basics

Topic Covered: ROS sensor_msgs

Part 3.1

Continution of Part 3

100DaysofROS series

By
M.N.Subhash
@malladisubhash

SENSOR_MSGS ROS MESSAGE PACKAGE

- This package contains messages for different sensors like cameras, rangefinders, joysticks, batteries. This makes it easier to read the data from the sensor and publish it to the required channel
- Different message types of this package are:
  1. **Battery State**: The message definition of this message type contains a lot of information about the battery some of the basic information provided is voltage, current, charge, and capacity.This data would be really useful as we can get to know everything about the battery by just a subscribing to the topic where this message is being published
  2. **LaserScan**: This message type is used when there is a laser type range finder in your robot, and provides all the basic data to get meaningful insights from it
  3. **Range**: This message type is used to get the range from the sensors, which are non-laser types, along with the range data there are some more useful data about the sensor being used
  4. **PointCloud**: This message is used in the navigation of the robot to direct the robot in a particular direction.

These are a few message types that are included in the sensor_msgs package, these few msg types are mainly used in the development of the autonomous navigation robot.

The range message type is used as a part of the **task 7**, which reads the data from the ultrasonic sensor and publishes all the data to the channel

Source: http://wiki.ros.org/sensor_msgs

The distance information calculated by the ultrasonic sensor is published to the topic "***/ultrasound_sensor***" by the publisher "***pub_range***" which accepts the range message type, along with it other information which is published to the channel is *radiation type* which is "ultrasound" in our case, and *field of view*, *min_range*, *max_range* and the distance information which is published as a *range*

The Arduino code is as follows

```
//This is the function which is called to publish the range_msg
void sensor_msg_init(sensor_msgs::Range &range_name, char *frame_id_name)
{
range_name.radiation_type = sensor_msgs::Range::ULTRASOUND;
range_name.header.frame_id = frame_id_name;
range_name.field_of_view = 0.26;
range_name.min_range = 0.0;
range_name.max_range = 2.0;
}

//This is used to create an instance of the Range and can be called multiple times depending on the number of sensors present
sensor_msgs::Range range_msg;

//This is how the publisher pub_range is create which publishes the message &range_msg to the /ultrasound_sensor topic
ros::Publisher pub_range("/ultrasound_sensor", &range_msg);
```

// The range data which is stored in the variable ***cm*** and along with the timestamp data is given to range_msg and published to the topic

```
range_msg.range = cm;
range_msg.header.stamp = nh.now();
pub_range.publish(&range_msg);
```

This explains the important snippets of the publisher code.

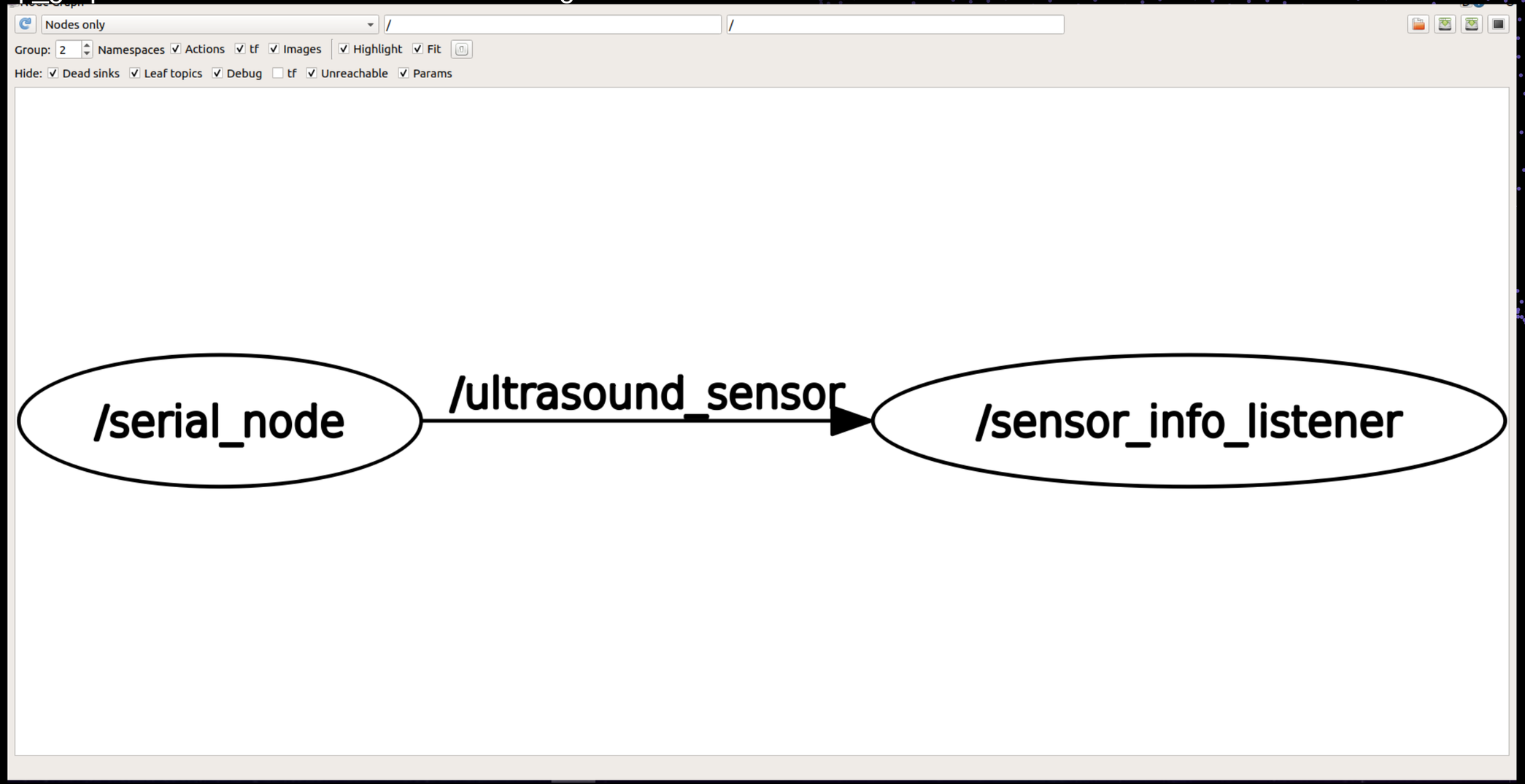
Now the PC subscribes to the topic /ultrasound and prints the range data is printed on to the monitor

```
def sensorInfoCallback(data):
rospy.loginfo('Distance reading from the sensor is : %f', data.range)

def sensorInfoListener():
rospy.init_node('sensor_info_listener', anonymous = False)
rospy.Subscriber('ultrasound_sensor',Range, sensorInfoCallback)
```

In this was sensor_range package is used to read the range data from the ultrasonic sensor.
link: https://github.com/malladi2610/100_days_of_ROS/tree/master/Day%2010%20-%2030

The rqt_graph of the task for better understanding



Here ***/serial_node*** is the node running on Arduino which is the publisher of the range data to the topic ***/ultrasound_sensor*** and this topic is subscribed by ***/sensor_info_listener***

About Me

I am currently pursuing my degree at Sir M Visvesvaraya Institute of Technology, Bangalore.
The domains I am interested in are Electronics and Robotics.
My interest in Robotics started when I was in my second year, Currently improving my skills in ROS and took a 100daysofROS challenge to focus mainly on the hardware implementation of ROS
I put regular updates of my progression of challenges on Twitter
As a part of the challenge, I will be documenting the important things and will be making a concise version of them.
Follow me to catch up with ROS and learn how to implement it on hardware.

LinkedIn profile: linkedin.com/in/malladisubhash
Twitter Handle: @mns2610
GitHub ROS repository link: https://github.com/malladi2610/100_days_of_ROS